



Федеральное государственное образовательное бюджетное учреждение высшего образования
«ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»
(Финансовый университет)

Кафедра информационных технологий
Факультет информационных технологий и анализа больших данных

Документ подписан усиленной неквалифицированной электронной подписью.

Организация: «ФИНАНСОВЫЙ УНИВЕРСИТЕТ ПРИ ПРАВИТЕЛЬСТВЕ
РОССИЙСКОЙ ФЕДЕРАЦИИ»

Сертификат: vu2tJNzen+5eEo9St/bneQBo96hfyUUX

Утверждено: Проректор по учебной и методической работе, Е.А. Каменева

Дата: 09.02.2026 г.

В.В. Андропов

Низкоуровневое программирование

Рабочая программа дисциплины

для студентов, обучающихся по направлению подготовки

01.03.02 - Прикладная математика и информатика,

Образовательная программа «Прикладное машинное обучение»

Профиль:

«Прикладное машинное обучение»

Рекомендовано

*Факультет информационных технологий и анализа больших данных
(протокол № 07 от 22.04.2026 г.)*

Одобрено

*Кафедра информационных технологий
(протокол № 04 от 13.04.2026 г.)*



Содержание

1. Наименование дисциплины	3
2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине	3
3. Место дисциплины в структуре образовательной программы	
4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся	5
5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий	6
5.1. Содержание дисциплины	6
5.2. Учебно – тематический план	9
5.3. Содержание семинаров, практических занятий	10
6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине	13
6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы	13
6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю	16
7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине	17
8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины	8
9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины	15
10. Методические указания для обучающихся по освоению дисциплины	27
11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)	28
12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине	28



1. Наименование дисциплины

Низкоуровневое программирование

2. Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине

Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
ПКП-7	Способность создавать прикладные программные компоненты и интегрировать в них методы анализа данных, машинного обучения и искусственного интеллекта	Владеет инструментальными средствами разработки прикладных программных продуктов: языками программирования, библиотеками, фреймворками	Знать: основы низкоуровневого программирования; архитектуру вычислительных систем (фон Нейман); систему команд микропроцессоров (x86, RISC-V); язык C как средство системного программирования; указатели и адреса памяти; побитовые операции; этапы создания исполняемого файла (компиляция, ассемблирование, компоновка); совместное использование C и ассемблера Уметь: разрабатывать программные компоненты на языке C с использованием указателей и побитовых операций; реализовывать низкоуровневые функции обработки данных; использовать gcc для сборки программ; применять ассемблерные вставки; обеспечивать корректную работу программных компонентов на уровне памяти и регистров
		Владеет архитектурными принципами и паттернами создания прикладных программных продуктов на различных платформах	Знать: архитектуру операционных систем; принципы организации памяти (стек, куча, сегменты); структуры данных ядра (таблицы процессов, дескрипторы); принципы построения загрузчика и ядра ОС; механизмы прерываний и системных вызовов; основы многозадачности и



Код компетенции	Наименование компетенции	Индикаторы достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции
			планирования процессов Уметь: проектировать программные компоненты с учетом архитектуры ОС; разрабатывать и модифицировать компоненты ядра (обработчики прерываний, системные вызовы); организовывать взаимодействие модулей ОС; учитывать особенности архитектуры процессора при разработке программных компонентов
		Демонстрирует понимание побочных процессов, сопровождающих прикладную разработку программного обеспечения в промышленных условиях и владеет соответствующим инструментарием	Знать: жизненный цикл программного обеспечения; процессы сборки, загрузки и выполнения программ; методы отладки низкоуровневых программ; инструменты анализа (GDB, Valgrind); принципы работы кэш-памяти; методы тестирования программных компонентов; особенности отладки операционных систем Уметь: выполнять отладку программ и компонентов ОС; анализировать состояние памяти и регистров процессора; использовать GDB для пошаговой отладки; применять Valgrind для выявления ошибок и анализа производительности; проводить тестирование и диагностику программных компонентов



3. Место дисциплины в структуре образовательной программы

Дисциплина «Низкоуровневое программирование» относится к «Модулю "Системное программирование"».

4. Объем дисциплины (модуля) в зачетных единицах и в академических часах с выделением объема аудиторной (лекции, семинары) и самостоятельной работы обучающихся

Очная форма обучения

Вид учебной работы по дисциплине	Всего (в з/е и часах)	Семестр 6 (в часах)
Общая трудоемкость дисциплины	3/108	108
Контактная работа – Аудиторные занятия	34	34
<i>Лекции</i>	<i>16</i>	<i>16</i>
<i>Семинары, практические занятия</i>	<i>18</i>	<i>18</i>
<i>Самостоятельная работа</i>	<i>74</i>	<i>74</i>
Вид текущего контроля	Контрольная работа	Контрольная работа
Вид промежуточной аттестации	Зачет	Зачет



5. Содержание дисциплины, структурированное по темам (разделам) дисциплины с указанием их объемов (в академических часах) и видов учебных занятий

5.1. Содержание дисциплины

Тема 1. Основные понятия и определения

Введение в низкоуровневое программирование. Цели и задачи дисциплины. Сведения об истории развития проблематики дисциплины. Эволюция вычислительных систем, архитектур ЭВМ и операционных систем. Уровни абстракции в программировании. Стандарты языков программирования (ANSI C, ISO C). Понятие низкоуровневого программирования. Файловые системы. Конфигурационная информация операционных систем. Сборка ядра Linux — как пример промышленной ОС для сопоставления с учебной ОС. Обзор проекта учебной операционной системы, разрабатываемой в рамках дисциплины.

Тема 2. Архитектура и системы команд современных микропроцессоров

Классическая архитектура вычислительных систем, предложенная Дж. фон Нейманом. Основные компоненты вычислительной системы: арифметико-логическое устройство, управляющее устройство, оперативная память, устройства ввода-вывода. Иерархия памяти. Регистровая память, кэш-память, оперативная память, внешняя память. Принципы организации и работы кэш-памяти. Современные структуры микропроцессоров. Архитектуры x86 и RISC-V. Регистры общего назначения и специальные регистры. Состояние процессора. Управление потоком команд. Прерывания и исключения. Системы команд микропроцессоров. Машинный код и ассемблер. Кодирование инструкций. Понятия загрузчика, ядра операционной системы и механизмов управления памятью и процессами при инициализации процессора в разрабатываемой операционной системе.

Тема 3. Введение в язык Си

Язык программирования Си как основной инструмент низкоуровневого и системного программирования. Структура программы на языке Си. Простые типы данных. Переменные и указатели. Работа с адресами памяти. Преобразование типов данных. Арифметические, логические и битовые операции. Операции сдвига и сравнения. Квалифицирующие типы и спецификаторы функций. Директивы препроцессора. Базовые операторы. Выражения и операции. Стандартные математические функции. Операторы выбора и цикла. Функции. Область видимости.



Передача параметров. Особенности использования языка Си при разработке ядра операционной системы. Ограничения стандартной библиотеки. Взаимодействие с ассемблерным кодом.

Тема 4. Составные типы данных

Составные типы данных: массивы. Объявление и инициализация массивов. Обращение к элементам массива. Упорядочивание массивов. Составные типы данных: структуры и объединения. Расположение элементов структур в оперативной памяти. Выравнивание и упаковка данных. Строки в языке С. Объявление и инициализация строк. Основные функции для работы со строками. Многомерные массивы. Инициализация и операции с многомерными массивами. Битовые поля и перечисления. Использование составных типов данных при описании структур ядра операционной системы: таблиц процессов, описателей памяти, системных структур.

Тема 5. Отладка низкоуровневых программ

Указатели и адреса памяти. Представление памяти процесса и ядра. Управление памятью. Области данных, текста программы, стека и кучи. Модели памяти. Динамическое выделение памяти. Особенности управления памятью в ядре операционной системы. Преобразования типов данных. Логические и битовые операции. Переменные окружения. Потоки ввода-вывода. Передача параметров. Этапы создания исполняемого файла: компиляция, ассемблирование, компоновка, загрузка. Отладка низкоуровневых программ. Отладка ядра операционной системы с использованием GDB. Анализ дампов памяти и регистров процессора. Поиск утечек памяти с использованием valgrind. Ассемблерные вставки в компиляторе gcc. Технологии дизассемблирования.

Тема 6. Разработка компонентов операционной системы

Архитектура операционной системы. Загрузчик и инициализация системы. Карта памяти. Переход в защищённый режим процессора. Реализация системных вызовов. Обработка прерываний и исключений. Таблицы дескрипторов. Процессы и потоки. Планирование процессов. Модификация и расширение планировщика процессов. Ввод-вывод и работа с устройствами. Практические навыки разработки компонентов ядра операционной системы.

Тема 7. Процессы, многозадачность и управление ресурсами

Модель процессов и потоков в операционных системах. Контексты выполнения. Переключение контекста. Алгоритмы планирования процессов. Кооперативная и вытесняющая многозадачность. Очереди процессов. Таймеры и обработка



прерываний таймера. Управление системными ресурсами. Синхронизация. Проблемы гонок данных и методы их предотвращения. Реализация и модификация подсистем управления процессами в рамках учебной операционной системы.

Тема 8. Отладка, тестирование учебной операционной системы

Методы отладки операционных систем. Использование эмуляторов и виртуальных машин. Отладка ядра с использованием GDB и удалённой отладки. Анализ загрузчика и карты памяти. Диагностика ошибок и сбоев на ранних этапах загрузки. Организация тестирования компонентов операционной системы. Поэтапное расширение функциональности ядра. Самостоятельная разработка, модификация и развитие учебной операционной системы.



5.2. Учебно – тематический план

Очная форма обучения

п/ п	Наименование тем (разделов) дисциплины	Трудоемкость в часах				
		Всего	Контактная работа - Аудиторная работа			Самостоятельная работа
			Общая, в т.ч.:	Лекции	Семинары, практические занятия	
1	Основные понятия и определения	8	4	2	2	4
2	Архитектура и системы команд современных микропроцессоров	14	4	2	2	10
3	Введение в язык Си	16	6	2	4	10
4	Составные типы данных	14	4	2	2	10
5	Отладка низкоуровневых программ	14	4	2	2	10
6	Разработка компонентов операционной системы	14	4	2	2	10
7	Процессы, многозадачность и управление ресурсами	14	4	2	2	10
8	Отладка, тестирование учебной операционной системы	14	4	2	2	10
В целом по дисциплине		108	34	16	18	74



5.3. Содержание семинаров, практических занятий

Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Основные понятия и определения	Понятие низкоуровневого программирования	Занятия в интерактивной форме в виде дискуссий и вводного практикума
Архитектура и системы команд современных микропроцессоров	Классическая архитектура ЭВМ, предложенная группой Дж. фон Неймана	Занятия в интерактивной форме в виде дискуссий
Введение в язык Си	Структура программы. Переменные, указатели, объявления в С. Преобразование типов данных в С. Логические, битовые, операции сдвига, сравнения в С. Директивы предпроцессора в С. Простые типы. Базовые операторы. Выражения и операции. Стандартные математические функции. Операторы выбора. Операторы цикла. Функции. Область видимости.	Интерактивная форма, практикум по решению задач по тематике занятия и коллективное обсуждение решений
Составные типы данных	Составные типы данных: массивы. Объявление и инициализация массивов. Обращение к элементам массива. Упорядочивание массивов. Составные типы данных: структуры и объединения. Расположение элементов структур в ОЗУ ЭВМ. Составные типы данных: строки. Объявление и инициализация строк. Функции для работы со строками. Составные типы данных: многомерные массивы. Инициализация	Занятия в интерактивной форме с элементами практического моделирования структур ядра



Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
	многомерных массивов. Операции с многомерными массивами.	
Отладка низкоуровневых программ	Представление памяти процесса и ядра. Сегменты данных, кода, стек и куча. Этапы создания исполняемого файла: компиляция, ассемблирование, компоновка, загрузка. Отладка низкоуровневых программ. Отладка ядра операционной системы с использованием GDB. Анализ регистров процессора и дампов памяти. Поиск утечек памяти с использованием valgrind. Ассемблерные вставки и дизассемблирование.	Интерактивная форма, практикум по отладке программ и коллективное обсуждение результатов
Разработка компонентов операционной системы	Архитектура операционной системы. Загрузчик и инициализация системы. Карта памяти. Переход в защищённый режим процессора. Реализация системных вызовов. Обработка прерываний и исключений. Таблицы дескрипторов. Процессы и потоки. Планирование процессов. Практическая разработка и модификация компонентов ядра учебной операционной системы.	Практические занятия с выполнением заданий в рамках разработки учебной операционной системы
Процессы, многозадачность и управление ресурсами	Модель процессов и потоков. Контекст выполнения. Переключение контекста. Алгоритмы планирования процессов. Кооперативная и вытесняющая многозадачность. Таймеры. Управление системными ресурсами. Синхронизация и проблемы гонок данных. Реализация и модификация подсистем управления процессами в учебной операционной системе.	Практикум с анализом и модификацией подсистем управления процессами



Наименование тем (разделов) дисциплины	Перечень вопросов для обсуждения на семинарах, практических занятиях	Формы проведения занятий
Отладка, тестирование учебной операционной системы	Методы отладки операционных систем. Использование эмуляторов и виртуальных машин. Удалённая отладка ядра с использованием GDB. Анализ загрузчика и карты памяти. Диагностика ошибок на ранних этапах загрузки. Организация тестирования компонентов операционной системы. Поэтапное расширение функциональности ядра.	Практические занятия, ориентированные на самостоятельную разработку, тестирование и отладку учебной операционной системы.



6. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

6.1. Перечень вопросов, отводимых на самостоятельное освоение дисциплины, формы внеаудиторной самостоятельной работы

Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Основные понятия и определения	История развития операционных систем. Эволюция архитектур ЭВМ. Уровни абстракции в программировании. Анализ примеров учебных ОС.	Изучение учебной и научной литературы, анализ материалов курса, подготовка конспектов.
Архитектура и системы команд современных микропроцессоров	Архитектуры x86 и RISC-V. Регистры процессора. Прерывания и исключения. Карта памяти при загрузке ОС.	Самостоятельное изучение литературы, анализ архитектурной документации, разбор примеров кода.
Введение в язык Си	Указатели памяти. Битовые операции. Ограничения стандартной библиотеки языка C при разработке ядра учебной ОС.	Выполнение индивидуальных заданий, написание и анализ программного кода.
Составные типы данных	Описание структур ядра: таблицы процессов, структуры управления памятью. Выравнивание и упаковка данных.	Самостоятельная разработка структур данных, анализ исходных текстов учебной ОС.
Отладка низкоуровневых программ	Этапы сборки исполняемых файлов. Использование GDB для анализа состояния процессора и памяти. Дизассемблирование.	Самостоятельная отладка программ, выполнение практических заданий.
Разработка компонентов операционной системы	Анализ загрузчика. Реализация и модификация системных вызовов. Работа с обработчиками прерываний.	Самостоятельная модификация исходного кода учебной ОС, выполнение практических заданий.
Процессы, многозадачность и управление ресурсами	Алгоритмы планирования процессов. Переключение контекста. Управление ресурсами учебной ОС.	Изучение теоретических материалов, анализ реализации планировщика процессов.



Наименование тем (разделов) дисциплины	Перечень вопросов, отводимых на самостоятельное освоение	Формы внеаудиторной самостоятельной работы
Отладка, тестирование учебной операционной системы	Методы отладки ОС. Использование эмуляторов и виртуальных машин. Анализ загрузки ОС.	Самостоятельная работа с эмуляторами, тестирование и расширение функциональности ОС.



6.2. Перечень вопросов, заданий, тем для подготовки к текущему контролю

Примерные задания контрольной работы

1. Проанализировать загрузчик учебной операционной системы. Описать этапы загрузки, карту памяти на этапе инициализации и назначение основных областей памяти. Указать, какие данные и структуры размещаются в каждой области.

2. Реализовать или модифицировать системный вызов в учебной операционной системе (например, вывод строки в консоль ядра либо получение системной информации). Описать механизм передачи управления от пользовательского кода к ядру.

3. Разработать обработчик программного или аппаратного прерывания. Описать процесс регистрации обработчика, сохранения и восстановления контекста процессора, а также порядок возврата из прерывания.

4. Произвести модификацию планировщика процессов учебной операционной системы: изменить алгоритм планирования либо добавить дополнительную очередь процессов. Обосновать выбранное решение и описать его влияние на выполнение процессов.

5. Реализовать минимальный механизм переключения контекста процессов. Описать сохраняемые и восстанавливаемые элементы контекста, а также условия, при которых происходит переключение.

6. Выполнить отладку ядра учебной операционной системы с использованием GDB. Проанализировать значения регистров процессора и состояние памяти при возникновении ошибки выполнения либо загрузки.

7. Произвести анализ дампа памяти или состояния процессора при сбое загрузки ядра. Определить возможную причину ошибки и предложить способ её устранения.

8. Адаптировать отдельный компонент учебной операционной системы (загрузчик, обработчик прерываний, подсистему управления памятью или процессами) для другой архитектуры процессора (x86 или RISC-V). Описать архитектурные различия и внесённые изменения.

9. Проанализировать и описать процесс перехода процессора в защищённый режим. Указать необходимые шаги и используемые аппаратные механизмы.

10. Реализовать и описать механизм инициализации одной из подсистем ядра операционной системы (управление памятью, обработка прерываний или управление процессами).

Критерии балльной оценки различных форм текущего контроля успеваемости содержатся в соответствующих методических рекомендациях



Кафедры информационных технологий Факультета информационных технологий и анализа больших данных.

Дополнительная информация



7. Фонд оценочных средств для проведения промежуточной аттестации обучающихся по дисциплине

Перечень компетенций с указанием индикаторов их достижения в процессе освоения образовательной программы содержится в разделе 2. «Перечень планируемых результатов освоения образовательной программы (перечень компетенций) с указанием индикаторов их достижения и планируемых результатов обучения по дисциплине».

Типовые контрольные задания или иные материалы, необходимые для оценки индикаторов достижения компетенций, умений и знаний

Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ПКП-7. Способность создавать прикладные программные компоненты и интегрировать в них методы анализа данных, машинного обучения и искусствен-	Владеет инструментальными средствами разработки прикладных программных продуктов: языками программирования, библиотеками, фреймворками	Знать: основы низкоуровневого программирования; архитектуру вычислительных систем (фон Нейман); систему команд микропроцессоров (x86, RISC-V); язык С как средство системного программирования; указатели и адреса памяти; побитовые операции;	реализовать программный модуль на языке С, выполняющий побитовую обработку данных (инверсия порядка битов в слове, упаковка логических переменных), проанализировать корректность работы на уровне представления в памяти; реализовать функции обработки байтов с использованием побитовых операций (логическое сложение/умножение с преобразованием порядка битов), обосновать используемые операции; разработать программу с использованием ассемблерных вставок, проанализировать взаимодействие С и ассемблерного кода; выполнить сборку программы с использованием gcc на разных этапах (компиляция, линковка), проанализировать структуру исполняемого файла.



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
ного интеллекта		этапы создания исполняемого файла (компиляция, ассемблирование, компоновка); совместное использование С и ассемблера Уметь: разрабатывать программные компоненты на языке С с использованием указателей и побитовых операций; реализовывать низкоуровневые функции обработки данных; использовать gcc для сборки программ; применять ассемблерные вставки; обеспечивать кор-	



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		ректную работу программных компонентов на уровне памяти и регистров	
	Владеет архитектурными принципами и паттернами создания прикладных программных продуктов на различных платформах	Знать: архитектуру операционных систем; принципы организации памяти (стек, куча, сегменты); структуры данных ядра (таблицы процессов, дескрипторы); принципы построения загрузчика и ядра ОС; механизмы прерываний и системных вызовов; основы многозадачности и планирования процессов Уметь: проектиро-	спроектировать архитектуру компонента учебной операционной системы (загрузчик, подсистема памяти или процессов), обосновать выбор структуры и взаимодействие компонентов; реализовать или модифицировать системный вызов, описать механизм передачи управления от пользовательского режима к ядру; разработать обработчик прерывания, проанализировать сохранение и восстановление контекста процессора; выполнить модификацию планировщика процессов, оценить влияние изменений на выполнение процессов; реализовать механизм переключения контекста, обосновать состав сохраняемых данных.



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		вать программные компоненты с учетом архитектуры ОС; разрабатывать и модифицировать компоненты ядра (обработчики прерываний, системные вызовы); организовывать взаимодействие модулей ОС; учитывать особенности архитектуры процессора при разработке программных компонентов	
	Демонстрирует понимание побочных процессов, сопровождающих прикладную разработку про-	Знать: жизненный цикл программного обеспечения; процессы сборки, загрузки и выполнения	выполнить отладку программы с использованием GDB, проанализировать состояние регистров и памяти при выполнении; провести анализ дампа памяти при сбое, выявить причину ошибки и предложить способ её устранения;



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
	граммного обеспечения в промышленных условиях и владеет соответствующим инструментарием	программ; методы отладки низкоуровневых программ; инструменты анализа (GDB, Valgrind); принципы работы кэш-памяти; методы тестирования программных компонентов; особенности отладки операционных систем Уметь: выполнять отладку программ и компонентов ОС; анализировать состояние памяти и регистров процессора; использовать GDB для пошаговой отладки; применять Valgrind	используя Valgrind (cachegrind), оценить количество промахов кэш-памяти и проанализировать причины; выполнить отладку загрузки учебной операционной системы, проанализировать карту памяти и этапы инициализации; организовать тестирование компонента ОС, оценить корректность и устойчивость его работы.



Наименование компетенции	Наименование индикаторов достижения компетенции	Результаты обучения (умения и знания), соотнесенные с индикаторами достижения компетенции	Типовые контрольные задания
		для выявления ошибок и анализа производительности; проводить тестирование и диагностику программных компонентов	



Примеры практико-ориентированных заданий

Задание:

Выполнить отладку ядра учебной операционной системы с использованием GDB:

1. проанализировать значения регистров процессора и состояние памяти при возникновении ошибки загрузки, либо некорректной работы ядра;
2. представить причины возникновения выявленных проблем и способов их устранения.

Примеры вопросов для подготовки к промежуточной аттестации

1. История развития вычислительных систем и операционных систем.
2. Понятие и особенности операционных систем класса OSDev.
3. Уровни абстракции в программировании.
4. Понятие низкоуровневого программирования.
5. Стандарты языков программирования ANSI C и ISO C.
6. Язык C как основной инструмент системного программирования.
7. Файловые системы.
8. Конфигурационная информация операционных систем.
9. Инструменты работы в Unix-подобных операционных системах.
10. Использование командной строки и текстового редактора vim.
11. Сборка и установка ядра операционной системы Linux.
12. Классическая архитектура ЭВМ, предложенная Дж. фон Нейманом.
13. Арифметико-логическое устройство и его функции.
14. Основное (оперативное) запоминающее устройство.
15. Иерархия памяти вычислительной системы.
16. Регистровая память и кэш-память.
17. Принципы организации и работы кэш-памяти.
18. Современные структуры микропроцессоров.
19. Архитектура x86: основные особенности.
20. Архитектура RISC-V: основные особенности.
21. Регистры общего назначения и специальные регистры процессора.
22. Состояние процессора и регистры флагов.
23. Управление потоком команд.
24. Прерывания и исключения.
25. Машинный код и ассемблер.
26. Кодирование инструкций.
27. Структура программы на языке C.



28. Простые типы данных языка С.
29. Переменные и указатели.
30. Работа с адресами памяти.
31. Преобразование типов данных в языке С.
32. Арифметические, логические и битовые операции.
33. Операции сдвига и сравнения.
34. Квалификаторы типов и спецификаторы функций.
35. Директивы препроцессора.
36. Функции и область видимости.
37. Передача параметров в функциях.
38. Массивы: объявление, инициализация, обращение к элементам.
39. Структуры и объединения.
40. Выравнивание и упаковка данных в памяти.
41. Строки в языке С и работа с ними.
42. Многомерные массивы.
43. Битовые поля и перечисления.
44. Использование составных типов данных в структурах ядра ОС.
45. Представление памяти процесса и ядра операционной системы.
46. Области данных, текста программы, стека и кучи.
47. Модели памяти.
48. Динамическое выделение памяти.
49. Особенности управления памятью в ядре ОС.
50. Этапы создания исполняемого файла: компиляция, ассемблирование.
51. Ассемблерные вставки в компиляторе gcc.
52. Технологии дизассемблирования.
53. Архитектура учебной операционной системы.
54. Загрузчик операционной системы и его назначение.
55. Анализ загрузчика и карты памяти при запуске ОС.
56. Переход процессора в защищённый режим.
57. Реализация системных вызовов.
58. Таблицы дескрипторов и обработка прерываний.
59. Процессы и потоки в операционной системе.
60. Контекст выполнения и переключение контекста.
61. Алгоритмы планирования процессов.
62. Модификация и расширение планировщика процессов.
63. Кооперативная и вытесняющая многозадачность.
64. Управление системными ресурсами.
65. Синхронизация и проблемы гонок данных.



- 66. Методы отладки низкоуровневых программ.
- 67. Отладка ядра операционной системы с использованием GDB.
- 68. Удалённая отладка ядра.
- 69. Анализ дампов памяти и регистров процессора.
- 70. Диагностика ошибок на ранних этапах загрузки ОС.
- 71. Поиск утечек памяти с использованием valgrind.
- 72. Профилирование программ.
- 73. Тестирование компонентов операционной системы.
- 74. Отличия разработки ОС под различные архитектуры процессоров.
- 75. Типичные ошибки при разработке операционных систем.
- 76. Требования к стабильности и надёжности ядра ОС.



8. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины

Основная литература:

1. Рацеев, С. М. Программирование на языке Си [Электронный ресурс] : учебное пособие для вузов / Рацеев С. М. 2-е изд., стер. Санкт-Петербург : Лань, 2023 332 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/351863> ISBN 978-5-507-47236-9. [БИК ID: RU-LAN-BOOK-351863]
2. Бунаков, П. Ю. Машинно-ориентированные языки программирования. Введение в ассемблер [Электронный ресурс] : учебное пособие для вузов / Бунаков П. Ю. Санкт-Петербург : Лань, 2023 144 с. Книга из коллекции Лань - Информатика <https://e.lanbook.com/book/302627> ISBN 978-5-507-45490-7. [БИК ID: RU-LAN-BOOK-302627]

Дополнительная литература:

1. Ларина, Татьяна Борисовна Низкоуровневые языки : Учебное пособие / Российский университет транспорта (МИИТ) Москва : Федеральное государственное автономное образовательное учреждение высшего образования «Российский университет транспорта», 2018 147 с. ВО - Бакалавриат <https://znanium.com/catalog/document?id=415617>. [БИК ID: RU\infra-m\znanium\bibl\1895286]

Нормативные правовые акты:

9. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Электронно-библиотечная система BOOK.RU <http://www.book.ru>
2. Электронная библиотека Финансового университета (ЭБ) <http://elib.fa.ru/> (<http://library.fa.ru/files/elibfa.pdf>)



10. Методические указания для обучающихся по освоению дисциплины

Самостоятельная работа обучающихся осуществляется в аудиторной и внеаудиторной формах. Организация самостоятельной работы обеспечивается календарно-тематическим планом изучения дисциплины. В указанном плане определяются тематика лекций и практических занятий, перечень вопросов и задания для самостоятельного освоения учебного материала.

При подготовке к лекционным занятиям рекомендуется предварительно ознакомиться с содержанием лекции по рекомендованным учебным и учебно-методическим пособиям, выделив наиболее сложные и проблемные вопросы. В ходе лекций следует вести конспектирование излагаемого материала. После лекционных занятий необходимо провести работу с конспектом: отредактировать записи, логически структурировать материал и оформить конспект. При оформлении целесообразно специально выделять названия тем, формулировки вопросов, основные определения, ключевые положения и примеры. Сделанные записи рекомендуется сверять с учебниками и учебными пособиями, а при возникновении расхождений или затруднений — обращаться за консультацией к преподавателю.

Студентам при подготовке контрольной работы следует использовать нормативные документы Финансового университета, Методические рекомендации по планированию и организации внеаудиторной самостоятельной работы студентов по образовательным программам бакалавриата и магистратуры в Финансовом университете, утвержденные приказом Финуниверситета от 11.05.2021 г. № 1040/о (см. сайт Финансового Университета: на главной странице раздел "Наш университет" → "Единая правовая база Финуниверситета" → "Организация учебного процесса" → "Нормативные документы по самостоятельной работе").



11. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень необходимого программного обеспечения и информационных справочных систем

Комплект лицензионного программного обеспечения:

1. Linux
2. Антивирус Kaspersky

Современные профессиональные базы данных и информационные справочные системы

1. eLIBRARY.RU
2. Электронная энциклопедия: <http://ru.wikipedia.org/wiki/Wiki>

Сертифицированные программные и аппаратные средства защиты информации

1. не предусмотрены

12. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине

1. **Компьютерный класс** для проведения учебных занятий, предусмотренных программой, в том числе групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, оснащенный оборудованием и техническими средствами обучения: мебель аудиторная (столы, стулья, доска аудиторная), персональные компьютеры, набор демонстрационного оборудования (проектор, экран)